

# Iterative graph cuts for image segmentation with a nonlinear statistical shape prior

Joshua C. Chang · Tom Chou

August 23, 2012

**Abstract** Shape-based regularization has proven to be a useful method for delineating objects from the noisy images encountered in many applications when one has prior knowledge of the shape of the targeted object. When a collection of possible shapes is available, the specification of a shape prior using kernel density estimation is a natural technique. Unfortunately, energy functionals arising from kernel density estimation are of a form that makes them impossible to directly minimize using graph cuts. Here, we show how one may recast the energy minimization problem into a form that is minimizable iteratively using graph cuts.

**Keywords** Image segmentation · MM · graph cuts · energy minimization · statistical shape prior · kernel density estimation

## 1 Introduction

Graph cuts provides an ingenious technique for image segmentation that relies on transforming the problem of energy minimization into the problem of determining the maximum flow or minimum cut on an edge-weighted graph. By using graph cuts, segmentations can be found efficiently in low-order polynomial time. When images are noisy, segmentation performed using only image data produces poor results. In these cases, regularization is needed.

Typically, when one is analyzing images, one has expectations for what he or she is looking for – people, birds, cars, cows, bacteria, whatever the object may be. In these cases it is natural to try to restrict segmentation to results that match the shape of the desired object. Within any class of objects there is often variability in shape. Such variability can arise even when dealing with shape-invariant objects, for example by changes in pose or perspective. Expression of shape knowledge probabilistically is a natural way of capturing this variability.

### 1.1 Related prior work

In Chang et al [4], we presented a technique for tracking an object whose boundary motion is modeled using the level-set method. Such objects do not retain any particular a-priori shape. We remarked that one could adapt our regularization method to the segmentation of objects that do retain their shapes. The purpose of this article is to explicitly present a method for performing such a task.

Our method is grounded in the theory of graph cuts-based image segmentation with shape-based regularization, where segmentation is performed using a-priori shape knowledge. The prior literature on this subject is vast. Briefly, we mention a few influential articles. One of the first successful forays into incorporating shape-based regularization into graph cuts was the work Slabaugh and Unal [22] who developed a method for segmentation assuming that the desired object is an ellipse. Allowing for more flexibility, Freedman and Zhang [12] provided a method to segment any particular shape based on the signed distance embedding. Some methods have concentrated on looking at shapes with certain characteristics such as compactness or local convexity [24, 9]. Other methods have treated the task of space specification as a statistical problem.

J.C. Chang  
UCLA Biomathematics  
BOX 951766, Room 5303 Life Sciences  
Los Angeles, CA 90095-1766  
E-mail: joshchang@ucla.edu

T. Chou  
UCLA Biomathematics and Mathematics  
E-mail: tomchou@ucla.edu

Statistical shape priors have taken many forms in the literature [18, 8, 27]. Two of the most common forms of shape representation are kernel-PCA and kernel-based level-set embeddings. Both methods learn a probability distribution for the target shape based on a set of training shapes or templates. They differ in how they use the training images. Kernel PCA-based methods project the training templates into spectral space and define probability distribution over features. Level-set-based methods define a distance between implicitly embedded shapes and define a probability distribution with respect to the distance.

One major disadvantage of the PCA-based approach is that the resulting probability distribution for the shape is Gaussian. Given a set of training templates, this approach will tend to favor shapes that look closer to the space averaged-shape, thereby potentially making certain training shapes improbable. This problem exists as well for exponential-family shape priors where the log-prior is a linear combination of the training templates. Combinations of shapes may not be valid shapes [7], and under this approach certain training templates may become improbable as well. Instead, we opt for a kernel-density based approach.

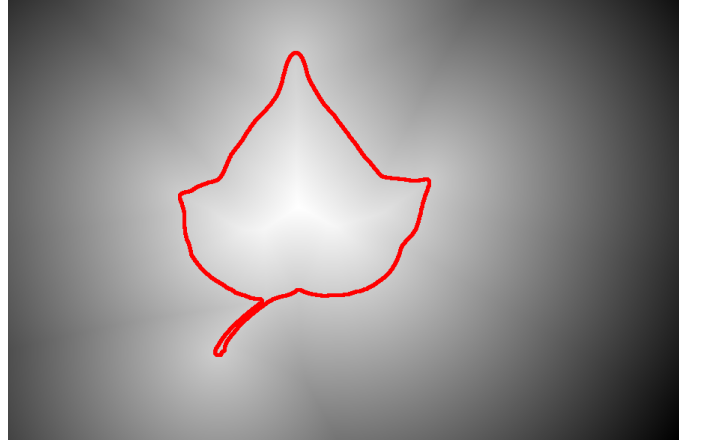
Our method here, and that given in Chang et al [4], can be described as a graph-cuts adaptation of the method of Cremers et al [6]. In their paper, they treated shape-prior specification as a density estimation problem and used kernel density estimation (KDE) to derive a prior from an ensemble of training shapes. The resulting energy functionals, being nonlinear, require some intervention to use in a graph-cuts framework.

In this manuscript, we make the main contribution of demonstrating the use of majorization-minimization (MM) for the iterative relaxation of a nonlinear energy functional using graph-cuts. The MM algorithm is a generalization of the well-known expectation-maximization (EM) algorithm. It is often useful for linearizing nonlinear objective functions. We use MM to find a surrogate minimization problem that is solved using graph-cuts. The use of graph cuts results in a computationally more efficient approach relative to that of using level set approaches [13].

## 2 Mathematical Method

### 2.1 Shapes

Signed-distance functions provide a handy tool to represent shapes mathematically. Shapes or regions  $\Omega \subset \mathbb{R}^d$  can be implicitly represented by a function  $\phi_\Omega(\mathbf{s}) : \mathbb{R}^d \rightarrow \mathbb{R}$ , which for every  $\mathbf{s} \in \mathbb{R}^d$  is the *signed Euclidean distance* from  $\mathbf{s}$  to the boundary of  $\Omega$ . In this paper we take the convention that  $\phi_\Omega(\mathbf{s})$  is positive if  $\mathbf{s} \in \Omega$ , and negative if  $\mathbf{s} \in \mathbb{R}^d \setminus \Omega$ .



**Fig. 1 Embedding of shapes using signed Euclidean distance functions.** A shape (pictured is a leaf) is embedded as a signed Euclidean distance function where values inside are the positive distance from the boundary, and values outside are negative distances from a boundary. The boundary of the leaf is the 0-level set of the function (red outline).

In many applications, since the shape of an object is only approximately known, it is advantageous to represent the knowledge of the shape probabilistically. We use a kernel density estimate of the distribution of possible shapes embedded as a collection of individual discrete level sets. Let us first denote  $\chi_\Omega$  the characteristic function for a region  $\Omega$ ,

$$\chi_\Omega(\mathbf{s}) = \begin{cases} 1, & \mathbf{s} \in \Omega; \\ 0, & \mathbf{s} \notin \Omega. \end{cases} \quad (1)$$

Then, for two shapes  $\Omega$  and  $\Lambda$ , embedded as signed distance functions  $\phi_\Omega$  and  $\phi_\Lambda$ , we introduce a pairwise shape energy

$$U_{\text{shape}}(\Omega, \Lambda) = \underbrace{\int_{\mathbb{R}^d} (\chi_\Omega(\mathbf{s}) - \chi_\Lambda(\mathbf{s}')) |\phi_\Lambda(\mathbf{s}')|^\lambda d\mathbf{s}}_{\text{mass mismatch}} + \underbrace{\oint_{\partial\Omega} |\phi_\Lambda(\mathbf{z}')|^\lambda d\mathbf{z}}_{\text{boundary mismatch}}. \quad (2)$$

This expression is a generalization of other shape energies seen in the literature, where commonly  $\lambda = 0, 1, 2$ . To make this energy robust to changes in scale, orientation, and location, a transformation  $\mathbf{s}' = \mathbb{T}(\mathbf{s})$  that maps points  $\mathbf{s}$  in the reference frame of  $\Omega$  to points  $\mathbf{s}'$  in the reference frame of  $\Lambda$  is needed. In Appendix A, we provide a method for finding rigid transformations of the form

$$\mathbb{T}(\mathbf{s}) = \alpha \mathbf{R}(\omega) (\mathbf{s} - \mathbf{c})$$

where  $\alpha$  is a scaling factor,  $\mathbf{R}(\omega)$  is a rotation matrix, and  $\mathbf{c}$  is a translation vector. These three parameters are chosen to minimize Eq 2 (See Appendix A for details on calculating these parameters). This method worked well for our examples; however, other shape alignment schemes may be used [16, 15, 26, 23, 1].

After transformation, we can use the energy in Eq 2 to assemble references shapes  $\{\Omega_j\}_{j=1}^J$  into a probability distribution by using a kernel of the form

$$K(\Omega, \Lambda) = \sqrt{\frac{\beta}{2\pi}} e^{-\beta U_{\text{shape}}(\Omega, \Lambda)}. \quad (3)$$

We can then represent a probability distribution over shapes,  $p(\Omega)$  as follows:

$$p(\Omega) \propto \sum_{j=1}^J w_j K(\Omega, \Omega_j), \quad (4)$$

where  $\{w_j\}_{j=1}^J$  are weighting coefficients. This representation of the prior is a kernel density estimate [21] (KDE) of the distribution of shapes. We set  $\beta$  to be a multiple of the following value:

$$\tau^2 = \left[ \sum_{j=1}^J w_j \min_{k \neq j} d(\Omega_k, \Omega_j) \right]^{-1}. \quad (5)$$

The parameter  $\beta$  can be considered a reciprocal “temperature,” or belief in the shape model implied by the given templates. If  $\beta = \tau^2$ , then the given templates give a good description of the possible shape space. In most applications, however, the number of templates available will be small and will not provide a complete picture of shape space. When the number of provided templates is small,  $\tau^2$  can in practice become rather small, leading to weak regularization. Setting  $\beta$  to a high multiple of  $\tau^2$  allows one to “boost” his or her belief in the available templates as a predictor of the segmented shape. Note that this expression of the shape prior is potentially multi-modal.

## 2.2 Generative image intensity model

The task is to identify an object  $\Omega$  in an image  $I : S \subset \mathbb{R}^d \rightarrow \mathbb{R}^q$ , where  $q$  refers to the dimension of the color space used. We model an image probabilistically with distributions of intensity values conditional on a parameter vector  $\theta$ , and region:

$$I(\mathbf{s}) \sim \begin{cases} p_{\Omega}(I|\theta) & \mathbf{s} \in \Omega \quad (\text{foreground}); \\ p_{\Delta}(I|\theta) & \mathbf{s} \in \Delta = S \setminus \Omega \quad (\text{background}). \end{cases} \quad (6)$$

Information about  $\theta$  and  $\Omega$  can be incorporated as prior probability distributions  $p(\theta|\Omega)$  and  $p(\Omega)$ . With these prior

distributions, we can write the joint posterior distribution  $p(\Omega, \theta|I) \propto p(I|\Omega, \theta)p(\theta|\Omega)p(\Omega) = e^{-U(\Omega, \theta)}$ , where  $U(\Omega, \theta)$  is an energy (not to be confused with  $U_{\text{shape}}$ ).

We wish to infer the segmentation  $\Omega$  by maximizing the posterior probability relative to  $\Omega$ . To this end, we will maximize the logarithm of the posterior, or equivalently, minimize the energy

$$\begin{aligned} U(\Omega, \theta) &= -\log \left[ \overbrace{p(I|\Omega, \theta)}^{\text{likelihood}} \overbrace{p(\theta|\Omega)p(\Omega)}^{\text{prior}} \right] \\ &= -\sum_{\mathbf{s} \in \Omega} \log p_{\Omega}(I(\mathbf{s})|\theta) - \sum_{\mathbf{s} \in \Delta} \log p_{\Delta}(I(\mathbf{s})|\theta) \\ &\quad - \underbrace{\log \sum_{j=1}^J w_j K(\Omega, \Omega_j)}_{\text{Eqs. 3, 4}} \\ &\quad - \log p(\theta|\Omega) + \text{const.} \end{aligned} \quad (7)$$

We take an iterative two-step approach to minimizing this energy. Given the segmentation estimate  $\Omega^{(n)}$  in the  $n$ -th step, we minimize the energy with respect to  $\theta$  [19].

Given  $\theta^{(n)}$ , we find the optimal  $\Omega^{(n+1)}$  by iteratively minimizing

$$\begin{aligned} Q(\Omega|\Omega^{(n)}) &= -\sum_{\mathbf{s} \in \Omega} \log p_{\Omega}(I(\mathbf{s})|\theta) - \sum_{\mathbf{s} \in \Delta} \log p_{\Delta}(I(\mathbf{s})|\theta) \\ &\quad + \underbrace{\frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} U_{\text{shape}}(\Omega, \Omega_j)}_{\text{majorization}}. \end{aligned} \quad (8)$$

In Eq 8, we have replaced logarithm-sum term of Eq. 8 with a linearized majorization term, by exploiting the concavity of the logarithm function. The majorization term defines a surrogate minimization problem that is more easily solved – in our case using graph cuts – albeit at the price of iteration. Such an algorithm is commonly known as a majorization-minimization or MM algorithm [14] (see Appendix B). Such an approach is generally possible for other energies that contain components of the form

$$f(J[\Omega]),$$

where the function  $f : \mathbb{R} \rightarrow \mathbb{R}$  is convex, and  $J$  is a functional dependent on  $\Omega$ .

## 2.3 Graph cuts for surrogate energy relaxation

Here, we describe minimization of the surrogate energy in Eq 8 using graph cuts. Graph cut methods have their grounding in combinatorial optimization theory, and are concerned with finding the minimum-cost cut in an undirected graph. A cut is a partition of a connected graph into two disconnected sets. The cost of a cut is the sum

of the edge weights along a cut, and a max-flow min-cut algorithm finds the cut with the lowest cost. To use graph cuts for image segmentation, we must express our energy function in terms of edge-weights on a graph. Following El Zehiry et al [10], we begin by expressing the energy given in Eq 8 as a function of the vertices  $V$  and edges  $\mathcal{E}$  of a graph  $\mathcal{G} = (V, \mathcal{E})$ :

$$U(\mathcal{G}) = \sum_{\mathbf{s} \in V} U_V(\mathbf{s}) + \sum_{(\mathbf{s}, \mathbf{u}) \in \mathcal{E}} U_{\mathcal{E}}(\mathbf{s}, \mathbf{u}).$$

In the graph that we construct, each pixel is assigned a node, and edges exist between nodes representing neighboring pixels (Fig 2). The neighbor edges are known as neighbor-links (n-links). Two special nodes called the source (foreground) and sink (background) are added, along with edges connecting these nodes to each pixel node (Fig 2). These edges are known as terminal-links (t-links).

We want to infer an unknown two-coloring on the nodes of the graph that represents inclusion of a node  $s$  into either the foreground set  $\Omega$ , or the background set  $\Delta$ . This inference involves splitting the graph into two parts (Fig 2), where a pixel's connection to the source represents inclusion into the foreground set.

Discretizing the shape divergence (Eq. 2) over an eight-connected neighbor graph yields

$$\begin{aligned} U_{\text{shape}}(\Omega, \Lambda) = & \sum_{\mathbf{s}} \overbrace{|\chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s})|}^{\text{indicator of label mismatch}} |\phi_{\Lambda}(\mathbf{s}')|^{\lambda} \\ & + \sum_{\mathbf{s}, \mathbf{u} \in \mathbf{N}} \overbrace{|(\chi_{\Omega}(\mathbf{s})(1 - \chi_{\Omega}(\mathbf{u})) + \chi_{\Omega}(\mathbf{u})(1 - \chi_{\Omega}(\mathbf{s})))|}^{\text{indicator of } \mathbf{s}, \mathbf{u} \text{ lying across boundary of } \Omega} \\ & \times \frac{\pi}{8\|\mathbf{s} - \mathbf{u}\|} \left| \phi_{\Lambda} \left( \frac{\mathbf{s}' + \mathbf{u}'}{2} \right) \right|^{\lambda}, \end{aligned} \quad (9)$$

where  $\mathbf{N}$  refers to the set of tuples of nodes that are neighboring in space. The distance function  $\phi_{\Lambda}^{\lambda}$  is taken to be the Euclidean distance embedding of  $\Lambda$ , transformed to the center, scale, and orientation of  $\Omega$ .

Using this expression, it is easy to see that

$$\begin{aligned} U_V(\mathbf{s}) = & -\chi_{\Omega}(\mathbf{s}) \log p_{\Omega}(I(\mathbf{s})|\boldsymbol{\theta}) - (1 - \chi_{\Omega}(\mathbf{s})) \log p_{\Delta}(I(\mathbf{s})|\boldsymbol{\theta}) \\ & + \frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) |\phi_{\Omega_j}(\mathbf{s})|^{\lambda} |\chi_{\Omega}(\mathbf{s}) - \chi_{\Omega_j}(\mathbf{s}')|}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} \end{aligned} \quad (10)$$

and

$$\begin{aligned} U_{\mathcal{E}}(\mathbf{s}, \mathbf{u}) = & \frac{\pi\beta}{16\|\mathbf{s} - \mathbf{u}\|} \left[ \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) \left| \phi_{\Omega_j} \left( \frac{\mathbf{s}' + \mathbf{u}'}{2} \right) \right|^{\lambda}}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} \right. \\ & \left. \times \left| \chi_{\Omega}(\mathbf{s})(1 - \chi_{\Omega}(\mathbf{u})) + \chi_{\Omega}(\mathbf{u})(1 - \chi_{\Omega}(\mathbf{s})) \right| \right]. \end{aligned} \quad (11)$$

The neighbor energy given in Eq 11 obeys the *submodularity property*, that is,

$$\begin{aligned} U_{\mathcal{E}}(\mathbf{s} \in \Omega, \mathbf{u} \notin \Omega) + U_{\mathcal{E}}(\mathbf{s} \notin \Omega, \mathbf{u} \in \Omega) \\ = & \frac{\pi\beta}{8\|\mathbf{s} - \mathbf{u}\|} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) \left| \phi_{\Omega_j} \left( \frac{\mathbf{s}' + \mathbf{u}'}{2} \right) \right|^{\lambda}}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} \\ \geq & U_{\mathcal{E}}(\mathbf{s} \in \Omega, \mathbf{u} \in \Omega) + U_{\mathcal{E}}(\mathbf{s} \notin \Omega, \mathbf{u} \notin \Omega) \\ = & 0. \end{aligned}$$

As a result, the energy (Eq 8) is minimizable with graph-cuts in polynomial time [2, 11].

To embed  $U_V(\mathbf{s})$  into the graph, we set the weights of the t-links between each pixel to the source to the following

$$\begin{aligned} w(\mathbf{s}, v_{\Omega}) = & -\log p_{\Delta}(I(\mathbf{s})|\boldsymbol{\theta}) \\ & + \frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) \chi_{\Omega_j}(\mathbf{s})}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} |\phi_{\Omega_j}(\mathbf{s}')|^{\lambda}, \end{aligned} \quad (12)$$

and the weights of the t-links between each pixel to the sink to the following

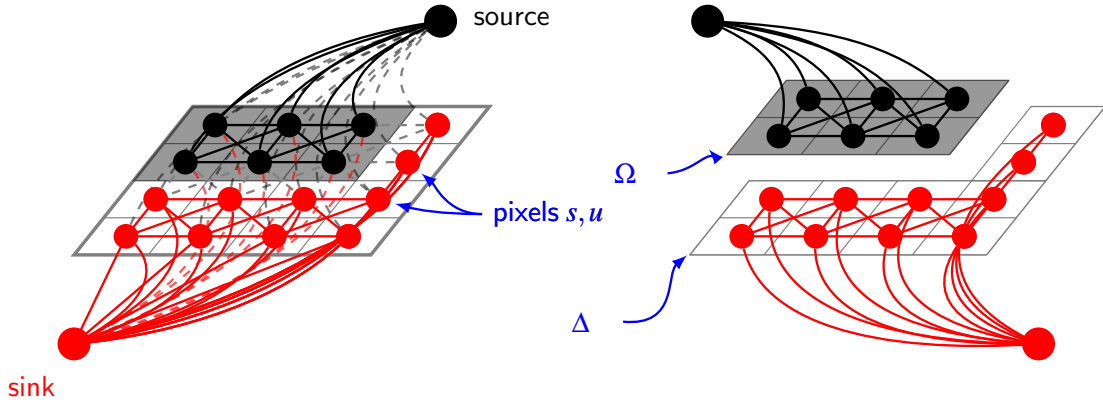
$$\begin{aligned} w(\mathbf{s}, v_{\Delta}) = & -\log p_{\Omega}(I(\mathbf{s})|\boldsymbol{\theta}) \\ & + \frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) (1 - \chi_{\Omega_j}(\mathbf{s}'))}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} |\phi_{\Omega_j}(\mathbf{s}')|^{\lambda}. \end{aligned} \quad (13)$$

The cutting of the edge from an  $\mathbf{s}$  to  $v_{\Omega}$  implies that  $\mathbf{s} \in \Delta$ , so it adds to the cost of the cut by the contribution of  $\mathbf{s}$  into the total energy as if  $\mathbf{s} \in \Delta$ . In other words, these weights can be interpreted as a pixel's strength of belonging to each region.

To embed  $U_{\mathcal{E}}$  into the graph, we set the n-links between pairwise neighboring pixels  $\mathbf{s}$  and  $\mathbf{u}$  to

$$\frac{\pi\beta}{16\|\mathbf{s} - \mathbf{u}\|} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j) \left| \phi_{\Omega_j} \left( \frac{\mathbf{s}' + \mathbf{u}'}{2} \right) \right|^{\lambda}}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)}. \quad (14)$$

Our surrogate energy is now minimized by finding the minimum cut of the graph. For details on how to perform this optimization, we refer the reader to Boykov and Kolmogorov [2].



**Fig. 2** *(left)* **Embedding of segmentation into a graph** In the graph-cuts framework, the problem of segmentation is transformed into the problem of finding the minimum cut in an edge-weighted graph. The energy to minimize is rewritten as a sum of atomic terms and a sum of neighbor-interaction terms. The edges are weighted such that the cut is the energy of the corresponding segmentation. *(right)* **Resulting segmentation.** The cutting of the graph yields two disconnected node sets where  $\Omega$  is the foreground and  $\Delta$  is the background.

### 3 Results

We have implemented our method in Java using the Fiji API [20] and tested it on the regularization of luminescence-based segmentation of a variety of objects in images (Fig 3). For comparison purposes, we have obtained results of segmentation using a shape-free global length penalty [10]. In these examples, we have used Laplacian statistics for  $p(I|\theta)$  and uninformative priors for  $\theta$  (by setting  $p(\theta) = 1$ ). The parameter  $\lambda$  was set to 2, and the inverse-temperature  $\beta$  was set to  $200\tau^2$ , except as otherwise noted.

The first images we processed were low-contrast noisy images of a van. A van (Dodge B2500) was driven across a bumpy road and videotaped as it entered and exited the camcorder's field of view. Snapshots of the recording were randomly extracted, showing the van in different regions

an of the image, under different orientations and difference scale factors. We manually constructed a shape prior for the van by hand-drawing five templates of a van in many poses. Shape-free segmentation of the van captured the dorsal aspect of the van, but was inaccurate in delineating its ventral features. Using the shape prior, we obtained an admissible result that is recognizable as a van. Although a van maintains a rigid structure, its shape in images can still be considered probabilistic due to uncertainty in pose.

We also processed pictures of leaves from the Caltech 101 image database<sup>1</sup>. In the given examples, background objects interfere with the segmentation. Without a shape prior, segmentations based on pixel intensities

<sup>1</sup> [http://www.vision.caltech.edu/Image\\_Datasets/leaves/](http://www.vision.caltech.edu/Image_Datasets/leaves/)

---

#### Algorithm 1 MM-Graphcut algorithm

---

Obtain initial guess of image intensity model parameters  $\theta^{(0)}$ .

Obtain initial guess of the segmentation  $\Omega^{(0)}$ , by segmentation without a shaper prior.

Align and rescale the shape prior templates to  $\Omega^{(0)}$ .

$n = 1$

**while**  $U_{\text{shape}}(\Omega^{(n)}, \Omega^{(n-1)}) > \text{tol}$  **do**

    Reweight the edges and perform max flow to estimate  $\Omega^{(n+1)}$

    Re-estimate the image intensity parameters  $\theta^{(n+1)}$  (as needed)

    Re-align the shape prior templates to the guess  $\Omega^{(n)}$  (as needed)

$n++$

**end while**

---

include these background distractors. To construct templates, we hand-traced six representative shapes for each of the two types of leaves. The regularized segmentations that we obtained were able to control for the background distractors.

In Fig 4, we segmented a five-pointed leaf using a prior composed of both three-tip and five-tip templates, using  $\beta = 600\tau^2$ . Although the mean shape of the hybrid-shape prior is a leaf with seven tips, the algorithm relaxed into the correct five-tip state.

## 4 Discussion

We have presented a method for regularization of segmentation using a shape prior that is learned through kernel-density-estimation. Using a majorization-minimization trick, we showed how one can minimize the resulting energy, that includes a non-linear kernel-density term, using graph-cuts. Although there have been previous studies on including statistical shape priors in graph-cuts segmentation [27, 5], these methods have used unimodal shape prior distributions where the input shape templates no longer retain their a-priori weights. Our shape-prior is a multi-modal kernel density estimator like the one used by Cremers et al [6], thereby enjoying all of the benefits outlined in Cremers et al [6].

The advantage of our method however is speed. Speed has become a big issue with the inception of the “big data” world. IT is also an issue in tracking applications, where many successive images need to be processed. Using an MM algorithm, we find surrogate energy functionals that can be minimized quickly using graph cuts. The surrogate energy functional (Eq 8) contains an iteratively re-weighted shape regularization term, where shapes that are closest to the current segmentation estimate are the most influential. In our examples, convergence of segmentation typically takes place in few ( $< 10$ ) iterations.

### 4.1 Choice of $\beta$ -multiplier

When the number of available templates are large, the kernel-density-estimated shape prior is a good representation of the shape space of the desired object. In this case,  $\beta \sim \tau^2$  can suffice. Yet, even in this case, there are situations where it is desirable to increase  $\beta$ . For example, in the event of likelihood mis-specification, increasing  $\beta$  can compensate for a poor likelihood model.

There are however caveats to increasing  $\beta$ . The major caveat is that if  $\beta$  becomes too large, the energy functional will tend to develop steep local minima about each shape template. Such a situation can prevent convergence of the algorithm. If a large  $\beta$  is necessary, then one suggestion is

to treat it as an annealing parameter and increase it only after a few iterations of the algorithm.

### 4.2 Computational considerations

Each iteration of the MM-algorithm is solved quickly using graph-cuts. Many efficient algorithms for this optimization exist, yielding excellent performance on modern hardware. There are many articles discussing the performance advantages of graph-cuts [3, 13]. For even more speed, there are GPU-accelerated approaches [25] and narrow-banding approaches [22, 17].

The minimization of the surrogate energy is not the bottleneck of our method. To unleash the full potential of this shape regularization method, one needs to optimize the processing steps that are in the periphery the graph cut optimization – the inference of the image intensity model, and the alignment of the templates.

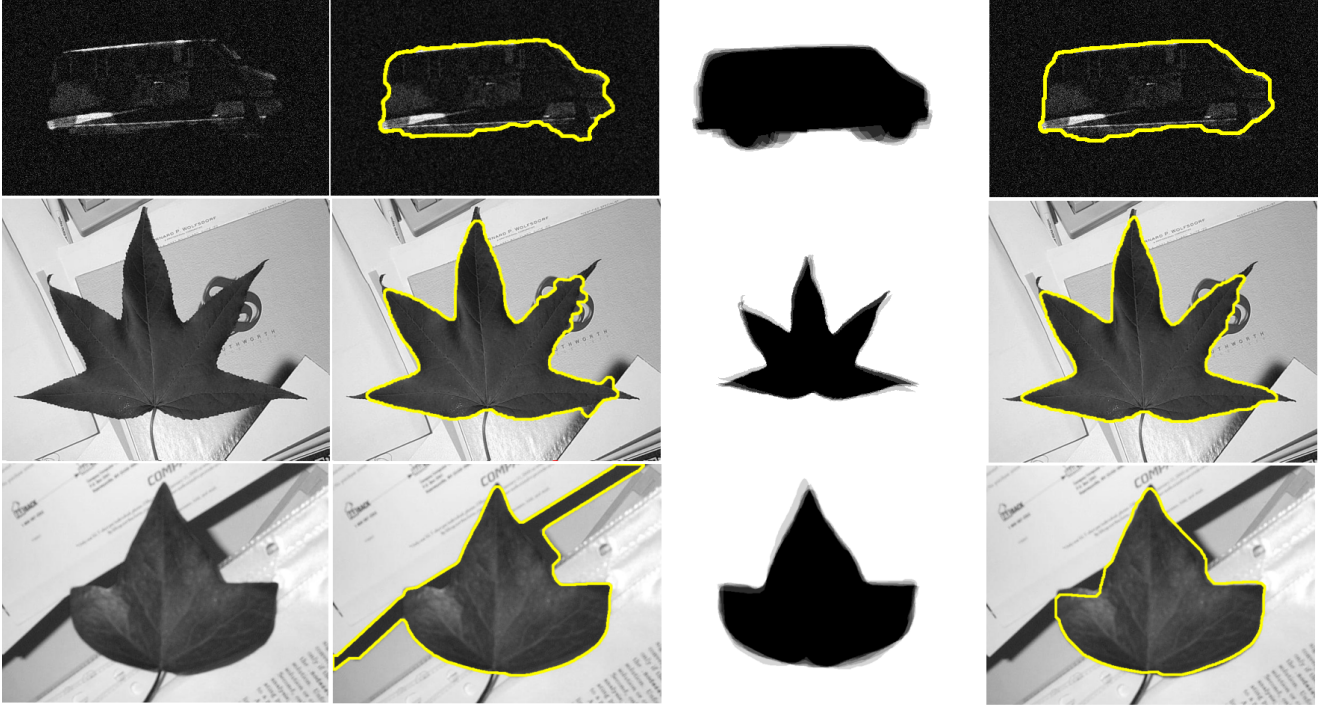
As far as we know, other methods of shape-regularized segmentation also require these processing steps. Fortunately, these operations can be easily parallelized. Furthermore, they need-not be repeated in every step of the algorithm. The image intensity parameters need only change if a significant proportion of the labels change. Similarly, the templates only need re-alignment if the segmentation has significantly changed. To improve performance, one may calculate the transformation needed for a single template, and use that transformation for the entire set of templates. One may also use a fast method such as that of Jiang and Tomasi [15] to provide an initial state for shape alignment.

### 4.3 Future directions

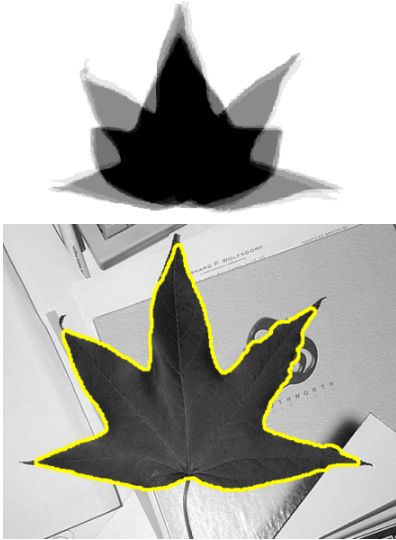
The MM algorithm is useful for linearizing objective functionals. In this paper, we have presented it as a technique for linearizing a nonlinear shape prior, yielding a linearized surrogate energy functional. It may in general be a useful tool for the graph-cut community in order to relax other useful nonlinear functionals. Finally, the MM and EM algorithms may have use in inference of more-robust likelihood models. In our opinion, it is an avenue worth investigating.

## 5 Acknowledgements

JC acknowledges support by grant number T32GM008185 from the National Institute of General Medical Sciences. JC and TC also acknowledge support from the the National Science Foundation through grants DMS-1032131 and DMS-1021818, and from the Army Research Office through grant 58386MA.



**Fig. 3 Shape-regularized segmentation** From left to right: original image, length-penalized segmentation, formation of a statistical shape prior using a collection of templates, shape-regularized segmentation. Segmentation of the original image of a van without shape information misses the wheels. A collection of templates taken together define a probabilistic shape prior. Such representation of shapes is useful since it can account for intrinsic variation in a class of shapes, as well as variations that result from changing pose. In these templates, silhouettes of a van are taken in different poses.



**Fig. 4 Segmentation using a hybrid shape-prior.** Using a combination of three and five-tip leaves, a “hybrid” shape prior is constructed. Although the average of the shape templates is a leaf with seven tips, the kernel-density shape prior model is able to settle into a shape with the proper configuration of five tips.

### A Shape alignment

In order to align the shape templates, we choose the affine transformation  $\mathbb{T}(\mathbf{s}) = \alpha \mathbf{R}(\boldsymbol{\omega})(\mathbf{s} - \mathbf{c})$  that minimizes the shape divergence energy

$$U_{\text{shape}}(\mathbb{T}(\mathbf{c}, \boldsymbol{\omega}, \alpha)) = \underbrace{\int_{\mathbb{R}^d} \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbb{T}(\mathbf{s})) \right)^2 \left| \phi_{\Lambda}(\mathbb{T}(\mathbf{s})) \right|^{\lambda} d\mathbf{s}}_{\text{mass mismatch}} + \underbrace{\int_{\mathbb{R}^d} \delta(\phi_{\Omega}(\mathbf{s})) \left| \phi_{\Lambda}(\mathbb{T}(\mathbf{s})) \right|^{\lambda} d\mathbf{s}}_{\text{boundary mismatch}}.$$

In this expression, we have rewritten the contour integral on  $\partial\Omega$  in terms of an integral over the zero-level set of  $\phi_{\Omega}$ . Here, we develop a local Newton-Raphson algorithm for finding the optimal transform. Let us denote the  $2 \cdot d$  column vector of transformation parameters  $\boldsymbol{\varphi} = [\alpha \ \mathbf{c} \ \boldsymbol{\omega}]$ . We estimate  $\boldsymbol{\varphi}$  using the iterative updates

$$\boldsymbol{\varphi}_{n+1} = \boldsymbol{\varphi}_n - [\mathbf{H}_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}_n)]^{-1} \nabla_{\boldsymbol{\varphi}} U_{\text{shape}}(\boldsymbol{\varphi}_n)$$



where

$$\nabla_{\boldsymbol{\varphi}} E = \begin{bmatrix} \frac{\partial}{\partial \alpha} & \nabla_{\mathbf{c}}^T & \nabla_{\boldsymbol{\omega}}^T \end{bmatrix}^T U_{\text{shape}}$$

and

$$\mathbf{H}_{\boldsymbol{\varphi}} = \begin{bmatrix} \frac{\partial^2}{\partial \alpha^2} & \nabla_{\mathbf{c}}^T \frac{\partial}{\partial \alpha} & \nabla_{\boldsymbol{\omega}}^T \frac{\partial}{\partial \alpha} \\ \nabla_{\mathbf{c}} \frac{\partial}{\partial \alpha} & \nabla_{\mathbf{c}} \nabla_{\mathbf{c}}^T & \nabla_{\boldsymbol{\omega}} \nabla_{\mathbf{c}}^T \\ \nabla_{\boldsymbol{\omega}} \frac{\partial}{\partial \alpha} & \nabla_{\boldsymbol{\omega}} \nabla_{\mathbf{c}}^T & \nabla_{\boldsymbol{\omega}} \nabla_{\boldsymbol{\omega}}^T \end{bmatrix} U_{\text{shape}}.$$

To populate the matrix, we need to calculate the gradients with respect to  $\alpha$ ,  $\mathbf{c}$ , and  $\boldsymbol{\omega}$ . For  $\lambda > 0$ , the first-order derivatives take the form

$$\begin{aligned} \nabla_{\boldsymbol{\varphi}_i} U_{\text{shape}} &= \lambda \int_{\mathbb{R}^d} \left[ \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right)^2 + \delta(\phi_{\Omega}(\mathbf{s})) \right] \\ &\quad \times \left| \phi_{\Lambda}(\mathbf{s}') \right|^{\lambda-1} \text{sgn}(\phi_{\Lambda}(\mathbf{s}')) \underbrace{\frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_i}}_{p \times d} \underbrace{\nabla \phi_{\Lambda}(\mathbf{s}')}_{d \times 1} d\mathbf{s} \end{aligned}$$

The sign function comes about from differentiation of the absolute value function in the distributional sense.

For  $\lambda \neq 1$ , the second-order derivatives take the form

$$\begin{aligned} \nabla_{\boldsymbol{\varphi}_j}^T \nabla_{\boldsymbol{\varphi}_i} U_{\text{shape}} &= \\ &\lambda(\lambda-1) \int_{\mathbb{R}^d} \left[ \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right)^2 + \delta(\phi_{\Omega}(\mathbf{s})) \right] \\ &\quad \cdot \left| \phi_{\Lambda}(\mathbf{s}') \right|^{\lambda-2} \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_j} \nabla \phi_{\Lambda}(\mathbf{s}') \nabla^T \phi_{\Lambda}(\mathbf{s}') \left( \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_i} \right)^T d\mathbf{s} \\ &+ 2\lambda \int_{\mathbb{R}^d} \delta(\phi_{\Lambda}(\mathbf{s}')) \left[ \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right)^2 + \delta(\phi_{\Omega}(\mathbf{s})) \right] \\ &\quad \cdot \left| \phi_{\Lambda}(\mathbf{s}') \right|^{\lambda-1} \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_i} \nabla \phi_{\Lambda}(\mathbf{s}') \nabla^T \phi_{\Lambda}(\mathbf{s}') \left( \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_j} \right)^T d\mathbf{s} \\ &+ \lambda \int_{\mathbb{R}^d} \left[ \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right)^2 + \delta(\phi_{\Omega}(\mathbf{s})) \right] \left| \phi_{\Lambda}(\mathbf{s}') \right|^{\lambda-1} \\ &\quad \cdot \text{sgn} \phi_{\Lambda}(\mathbf{s}') \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_i} \nabla \nabla \phi_{\Lambda}(\mathbf{s}') \left( \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_j} \right)^T d\mathbf{s} \\ &+ \lambda \int_{\mathbb{R}^d} \left[ \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right)^2 + \delta(\phi_{\Omega}(\mathbf{s})) \right] \\ &\quad \cdot \left| \phi_{\Lambda}(\mathbf{s}') \right|^{\lambda-1} \text{sgn} \phi_{\Lambda}(\mathbf{s}') \underbrace{\frac{\partial^2 \mathbb{T}}{\partial \boldsymbol{\varphi}_j^T \partial \boldsymbol{\varphi}_i}}_{p \times p \times d} : \nabla \phi_{\Lambda}(\mathbf{s}') d\mathbf{s}. \end{aligned}$$

If  $\lambda = 1$ , the first term is zero and the following term is added

$$\begin{aligned} &- 2\lambda \int_{\mathbb{R}^d} \delta(\phi_{\Lambda}(\mathbf{s}')) \left( \chi_{\Omega}(\mathbf{s}) - \chi_{\Lambda}(\mathbf{s}') \right) \text{sgn}(\phi_{\Lambda}(\mathbf{s}')) \\ &\quad \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_j} \nabla \phi_{\Lambda}(\mathbf{s}') \left( \frac{\partial \mathbb{T}}{\partial \boldsymbol{\varphi}_i} \nabla \phi_{\Lambda}(\mathbf{s}') \right)^T d\mathbf{s}. \end{aligned}$$

The contour integrals in these expressions can be calculated using a regularized version of the Dirac delta function such as

$$\delta_{\epsilon}(x) = \frac{1}{\pi} \frac{\epsilon}{\epsilon^2 + x^2} \quad \text{for small } \epsilon.$$

Similarly, the characteristic function can be interpreted as the Heaviside function acting on the signed-distance shape embedding, which can be approximated using an approximation of the Heaviside function such as

$$H_{\epsilon}(x) = \frac{1}{2} + \frac{1}{\pi} \arctan \frac{x}{\epsilon} \quad \text{for small } \epsilon.$$

The nonzero transformation derivatives are as follows

$$\begin{aligned} \frac{\partial \mathbb{T}}{\partial \mathbf{c}} &= -\alpha \mathbf{R} \\ \frac{\partial \mathbb{T}}{\partial \alpha} &= \mathbf{R}(\mathbf{s} - \mathbf{c}) \\ \frac{\partial \mathbb{T}}{\partial \boldsymbol{\omega}} &= \alpha \frac{\partial \mathbf{R}}{\partial \boldsymbol{\omega}}(\mathbf{s} - \mathbf{c}) \\ \frac{\partial^2 \mathbb{T}}{\partial \boldsymbol{\omega} \partial \mathbf{c}} &= -\alpha \frac{\partial \mathbf{R}}{\partial \boldsymbol{\omega}} \\ \frac{\partial^2 \mathbb{T}}{\partial \alpha \partial \boldsymbol{\omega}} &= \frac{\partial \mathbf{R}}{\partial \boldsymbol{\omega}}(\mathbf{s} - \mathbf{c}) \\ \frac{\partial^2 \mathbb{T}}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}} &= \alpha \frac{\partial^2 \mathbf{R}}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}}(\mathbf{s} - \mathbf{c}) \\ \frac{\partial^2 \mathbb{T}}{\partial \alpha \partial \mathbf{c}} &= -\mathbf{R}. \end{aligned}$$

## B MM algorithm for iterative graph cuts

The shape term  $\log \sum w_j K(\Omega, \Omega_j)$  can make minimization of the energy difficult, since its formulation involves a sum within a logarithm, making the energy functional nonlinear with respect to the labeling of pixels (into background and foreground) in the image.

To linearize the shape contribution, we will derive a surrogate function with separated terms. A function  $f(x|x_k)$ , with fixed  $x_k$ , is said to majorize a function  $g(x)$  at  $x_k$  if the following holds [14]

$$\begin{aligned} g(x) &\leq f(x|x_k) \\ f(x_k) &= g(x_k|x_k). \end{aligned}$$

We wish to perform iterative inference by finding a sequence of segmentations  $\Omega^{(n+1)} = \arg \min_{\Omega} Q(\Omega|\Omega^{(n)})$ , where  $Q(\Omega|\Omega^{(n)})$  majorizes Eq 7. By the descent property of the MM algorithm [14], this sequence converges to a local minimum.

For any convex function  $f(x)$ , the following holds [14]

$$f\left(\sum_i \alpha_i t_i\right) \leq \sum_i \alpha_i f(t_i).$$



Noting that  $-\log(\cdot)$  is convex, we have

$$\begin{aligned}
& \overbrace{-\log \sum_{j=1}^J w_j K(\Omega, \Omega_j)}^{\text{shape kernel density}} \leq \\
& -\sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} \\
& \quad \times \log \left[ \frac{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)}{w_j K(\Omega^{(n)}, \Omega_j)} w_j K(\Omega, \Omega_j) \right] \\
& = -\sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} \log K(\Omega, \Omega_j) \\
& \quad + \text{const} \\
& = \underbrace{\frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} U_{\text{shape}}(\Omega, \Omega_j)}_{\text{separated log shape kernel density}} \\
& \quad + \text{const.} \tag{15}
\end{aligned}$$

Since Eq 15 majorizes the log-kernel density, we can minimize our original energy by iteratively minimizing

$$\begin{aligned}
Q(\Omega | \Omega^{(n)}) &= -\sum_{s \in \Omega} \log p_{\Omega}(I(s) | \theta) - \sum_{s \in \Delta} \log p_{\Delta}(I(s) | \theta) \\
&\quad - \log p(\theta | \Omega) \\
&\quad + \frac{\beta}{2} \sum_{j=1}^J \frac{w_j K(\Omega^{(n)}, \Omega_j)}{\sum_{k=1}^J w_k K(\Omega^{(n)}, \Omega_k)} U_{\text{shape}}(\Omega, \Omega_j). \tag{16}
\end{aligned}$$

Since the distance function can be written as a sum over the vertices, so can Eq 16. As a result, it is possible to minimize Eq 16 within the graph cuts framework.

## References

1. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 24(4):509–522
2. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 26(9):1124–1137
3. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 23(11):1222–1239
4. Chang J, Chou T, Brennan K (2012) Tracking monotonically advancing boundaries in image sequences using graph cuts and recursive kernel shape priors. *Medical Imaging*, IEEE Transactions on 13(5):1008–1020
5. Cremers D, Grady L (2006) Statistical priors for efficient combinatorial optimization via graph cuts. *Computer Vision–ECCV 2006* pp 263–274
6. Cremers D, Osher S, Soatto S (2006) Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International journal of Computer Vision* 69(3):335–351
7. Cremers D, Schmidt F, Barthel F (2008) Shape priors in variational image segmentation: Convexity, lipschitz continuity and globally optimal solutions. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, pp 1–6
8. Dambreville S, Rathi Y, Tannenbaum A (2008) A framework for image segmentation using shape models and kernel space shape priors. *IEEE transactions on pattern analysis and machine intelligence* pp 1385–1399
9. Das P, Veksler O, Zavadsky V, Boykov Y (2009) Semi-automatic segmentation with compact shape prior. *Image and Vision Computing* 27(1-2):206–219
10. El Zehiry N, Xu S, Sahoo P, Elmaghraby A (2007) Graph cut optimization for the Mumford-Shah model. In: *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*, ACTA Press, pp 182–187
11. Freedman D, Drineas P (2005) Energy minimization via graph cuts: Settling what is possible. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol 2, pp 939–946
12. Freedman D, Zhang T (2005) Interactive graph cut based segmentation with shape priors. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, vol 1, pp 755–762
13. Grady L, Alvino C (2009) The piecewise smooth mumford-shah functional on an arbitrary graph. *Image Processing*, IEEE Transactions on 18(11):2547–2561
14. Hunter D, Lange K (2004) A tutorial on MM algorithms. *The American Statistician* 58(1):30–37
15. Jiang T, Tomasi C (2008) Robust shape normalization based on implicit representations. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, IEEE, pp 1–4
16. Lee S, Park J (1994) Nonlinear shape normalization methods for the recognition of large-set handwritten characters. *Pattern Recognition* 27(7):895–902
17. Lombaert H, Sun Y, Grady L, Xu C (2005) A multi-level banded graph cuts method for fast image seg-

- mentation. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, IEEE, vol 1, pp 259–265
18. Malcolm J, Rathi Y, Tannenbaum A (2007) Graph cut segmentation with nonlinear shape priors. In: Image Processing, 2007. ICIP 2007. IEEE International Conference on, IEEE, vol 4
19. O’Hagan A, Forster J, Kendall M (2004) Bayesian inference. Arnold
20. Schindelin J (2008) Fiji is just ImageJ – batteries included. In: Proceedings of the ImageJ User and Developer Conference, Luxembourg
21. Silverman B (1986) Density estimation for statistics and data analysis, vol 26. Chapman & Hall/CRC
22. Slabaugh G, Unal G (2005) Graph cuts segmentation using an elliptical shape prior. In: Image Processing, 2005. ICIP 2005. IEEE International Conference on, IEEE, vol 2
23. Tabbone S, Wendlings L (2003) Binary shape normalization using the radon transform. In: Discrete Geometry for Computer Imagery, Springer, pp 184–193
24. Veksler O (2008) Star shape prior for graph-cut image segmentation. Computer Vision–ECCV 2008 pp 454–467
25. Vineet V, Narayanan P (2008) Cuda cuts: Fast graph cuts on the gpu. In: Computer Vision and Pattern Recognition Workshops, 2008. CVPRW’08. IEEE Computer Society Conference on, IEEE, pp 1–8
26. Viola P, Wells III W (1997) Alignment by maximization of mutual information. International journal of computer vision 24(2):137–154
27. Zhu-Jacquot J, Zabih R (2007) Graph cuts segmentation with statistical shape priors for medical images. In: Signal-Image Technologies and Internet-Based System, 2007. SITIS’07. Third International IEEE Conference on, IEEE, pp 631–635